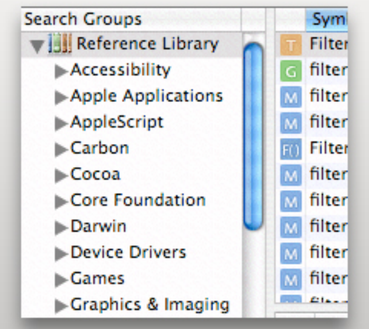
What's New in Xcode 2.1

A Guide to Features and Enhancements

Session 401

These are confidential sessions—please refrain from filming or taking pictures.



Development Tools

What's New in Xcode 2.1

A Guide to Features and Enhancements

Chris Espinosa
DT Productivity
cde@apple.com

http://developer.apple.com/wwdc2005/

What You'll Learn

- A Quick Recap of Major 2.0 Features
- New Key Features of Xcode 2.1
  - Build Configurations
  - Breakpoint Actions, Conditions, and Watchpoints
  - Integrated Unit Testing
  - Project File Format Changes
- Xcode Support for Universal Binaries

Introduction: Xcode 2.1

- Replaces Tiger's Xcode 2.0
- Runs on Mac OS X 10.4 ("Tiger") and later
  - Both PowerPC and Intel
- Develops for Mac OS X 10.2 and later
- Includes:
  - Mac OS X 10.4 universal SDK
  - Features to support large project management
  - Minor enhancements to improve your productivity
  - Dozens of bug fixes
- New project file format and extension
  - Upward (but not backward-) compatible with Xcode 2.0 and earlier

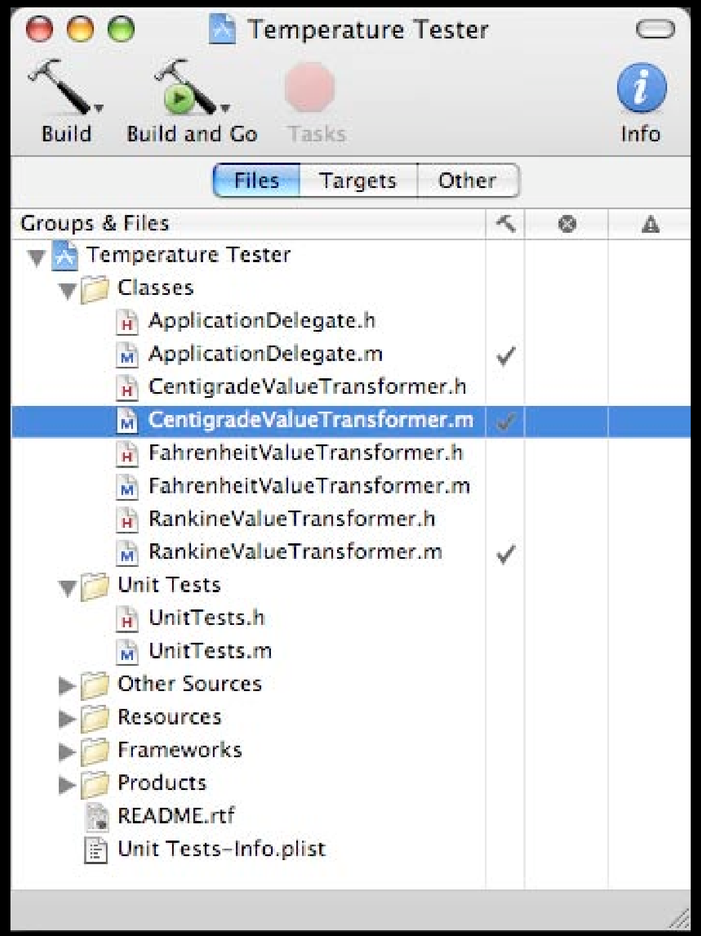
Highlights of Xcode 2.0: a Recap

- New UI layouts
- Design tools
- “Favorites” bar
- Code completion improvements
- New documentation
- Improved AppleScriptability

Layouts

Use your preferred workflow

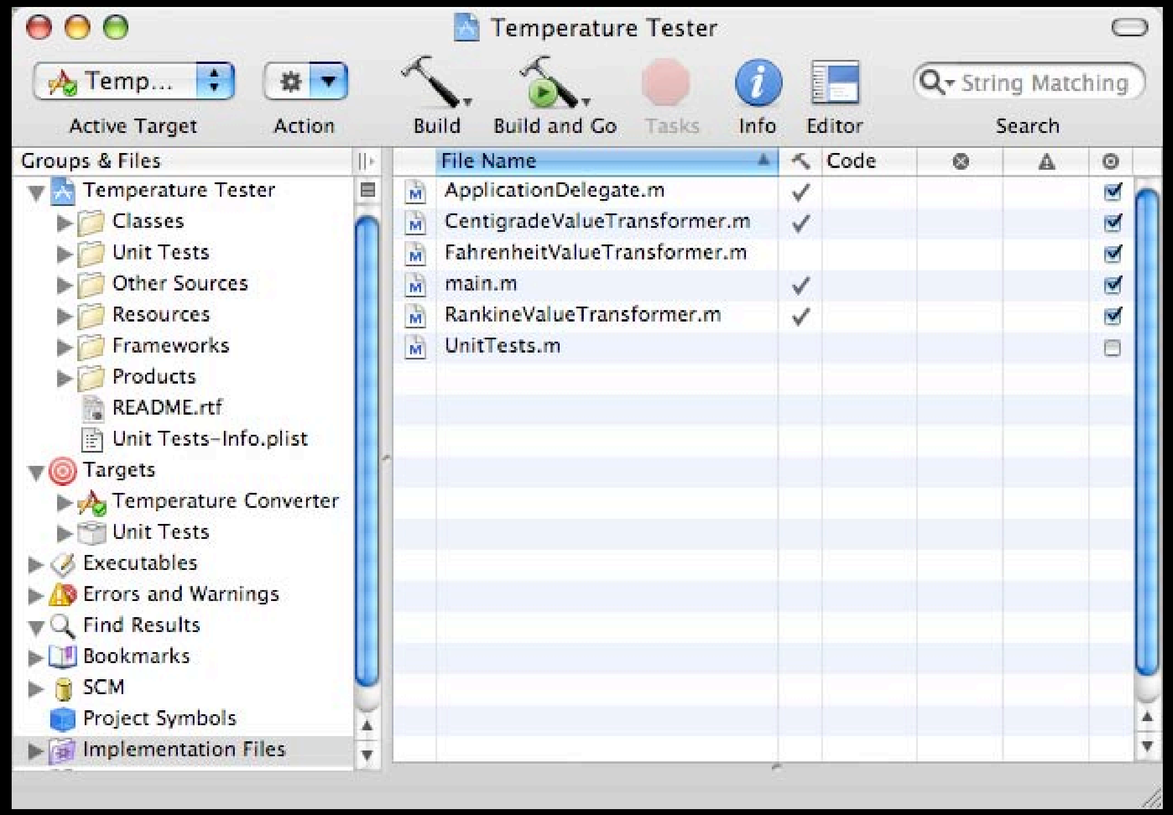
- “Condensed” is CodeWarrior-like



Layouts

Use your preferred workflow

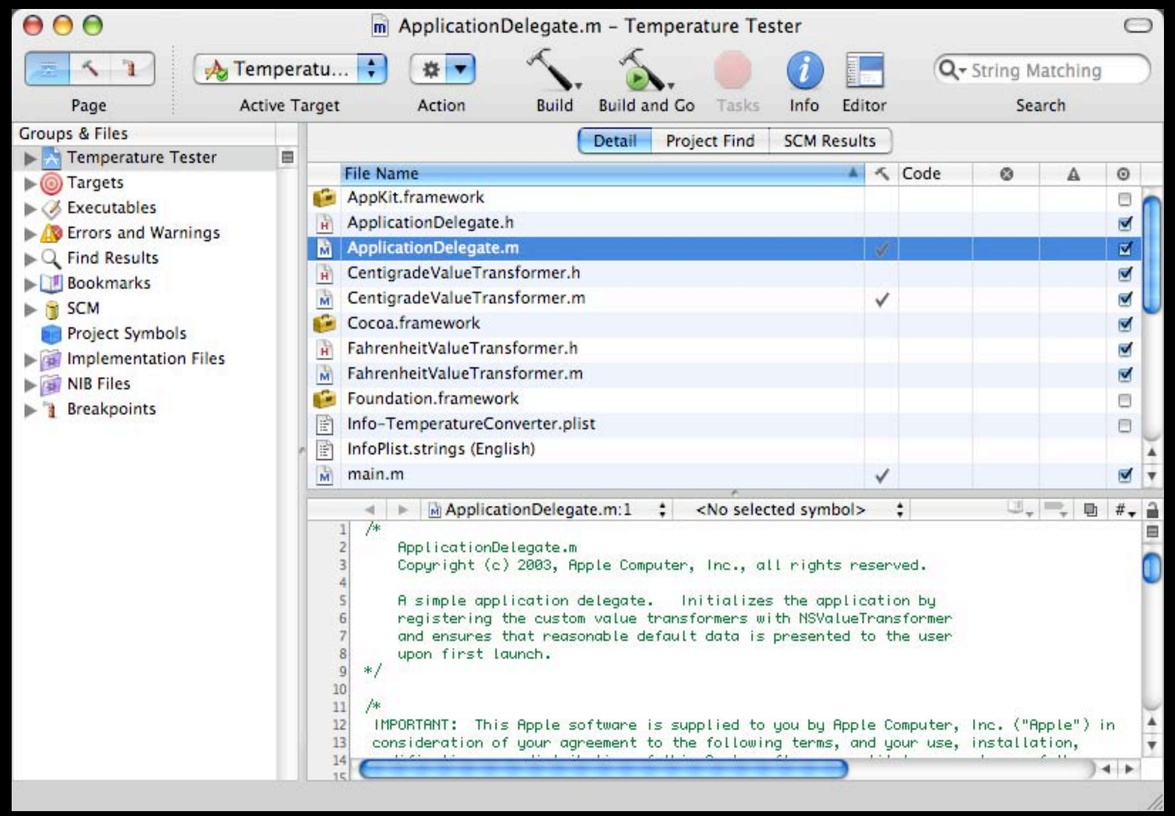
- “Condensed” is CodeWarrior-like
- “Default” is Xcode native



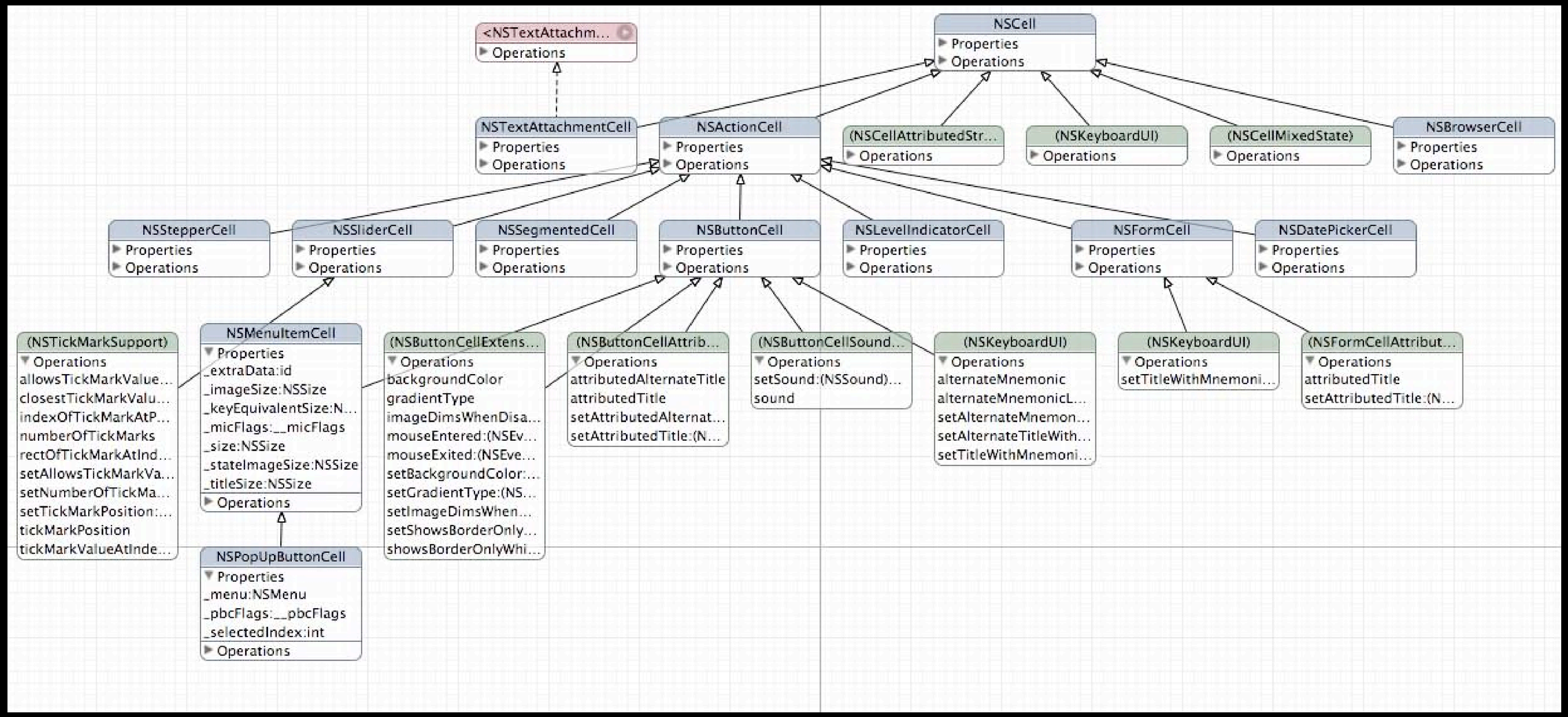
Layouts

Use your preferred workflow

- “Condensed” is CodeWarrior-like
- “Default” is Xcode native
- “All in one” is Visual Studio-like



Design Tools



Code Completion

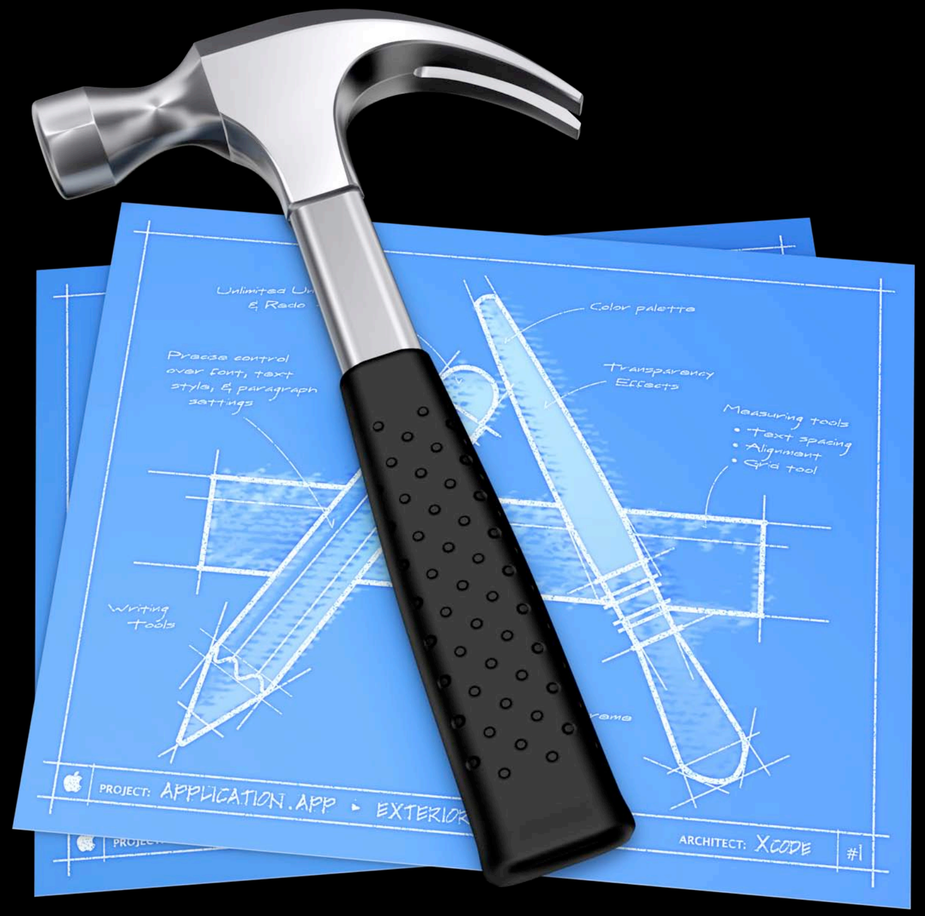
```
53 - (void)stockChanged:(NSString *)stockSymbol newPrice:(double)newPrice {
54     NSString *newLine = [NSString stringWithFormat:@"%@ new price: $%.2f\n", stockSymbol,
                            newPrice];
55
56 }
57
58
```

Xcode 2.1

http://developer.apple.com/wwdc2005/

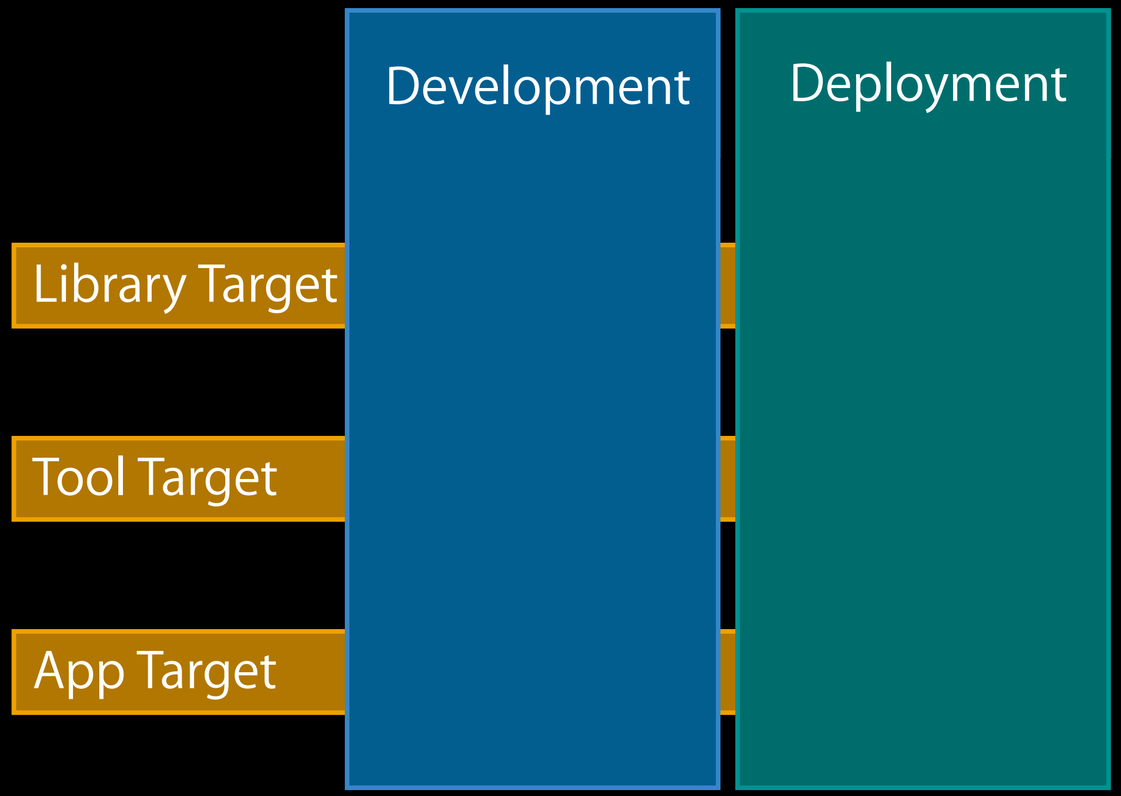
Xcode 2.1 Motivations

- Support the transition to universal binaries
  - Build configurations
  - Unit testing
  - New project file format
- Improve liveability
  - Advanced debugging
  - Many UI changes



Build Styles → Build Configurations

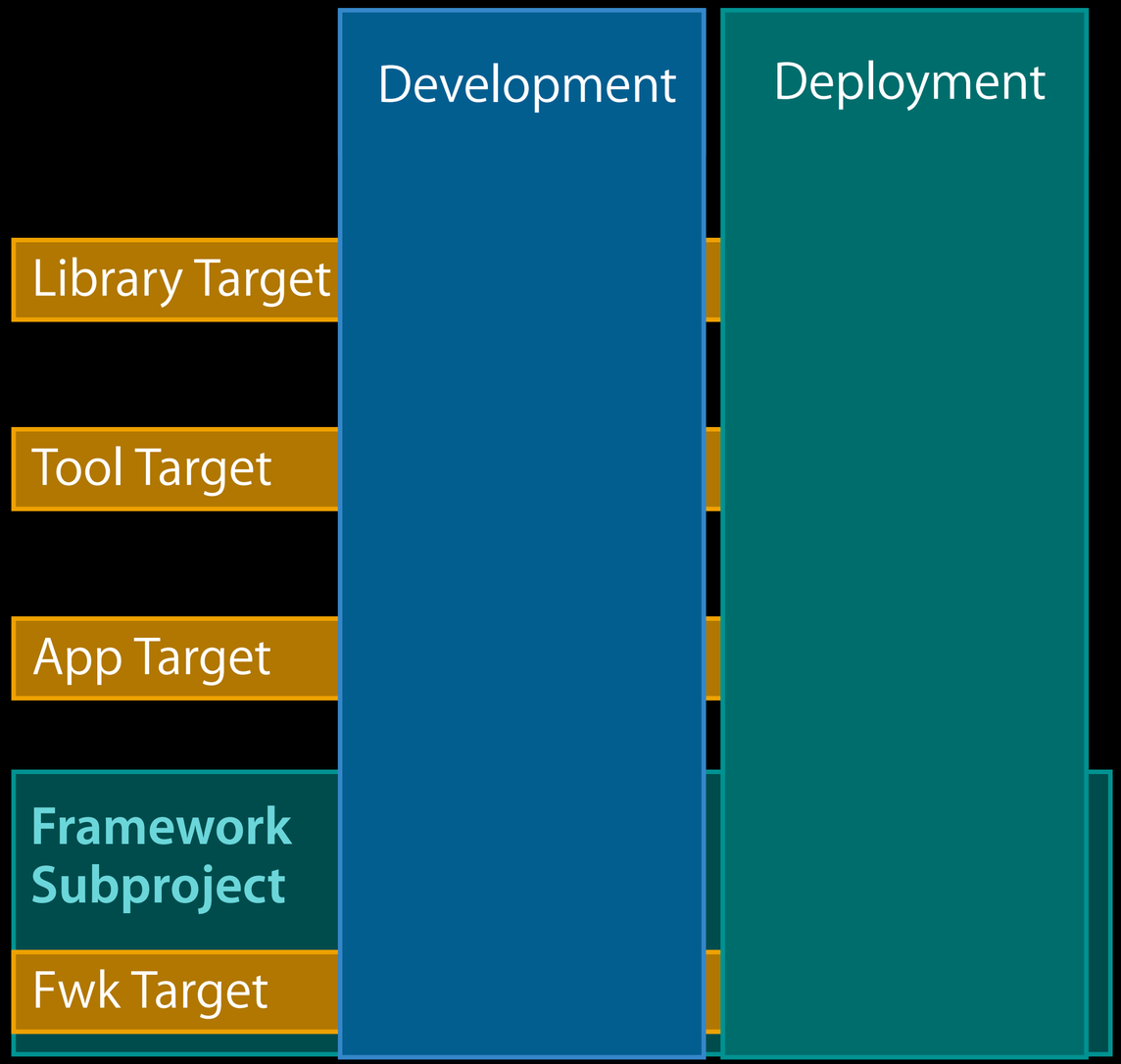
http://developer.apple.com/wwdc2005/



Build Styles

A product of build style and target

- Build styles imposed on all targets
- Style settings override target settings



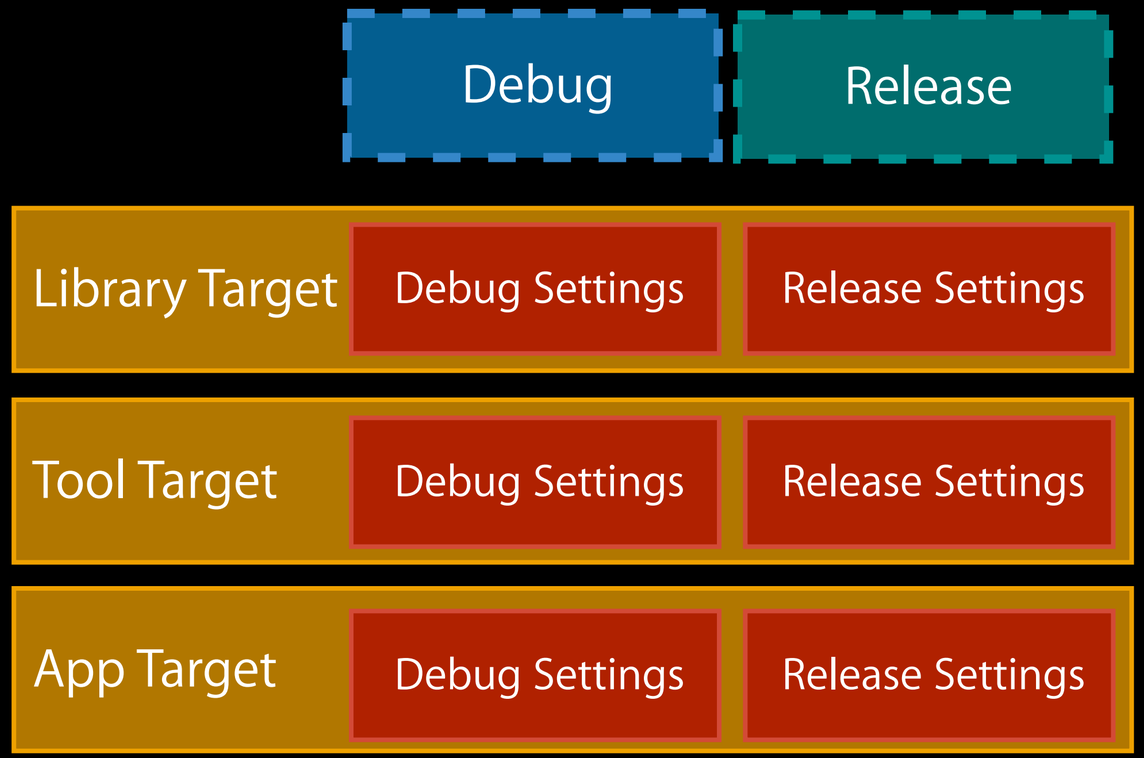
Build Styles

A product of build style and target

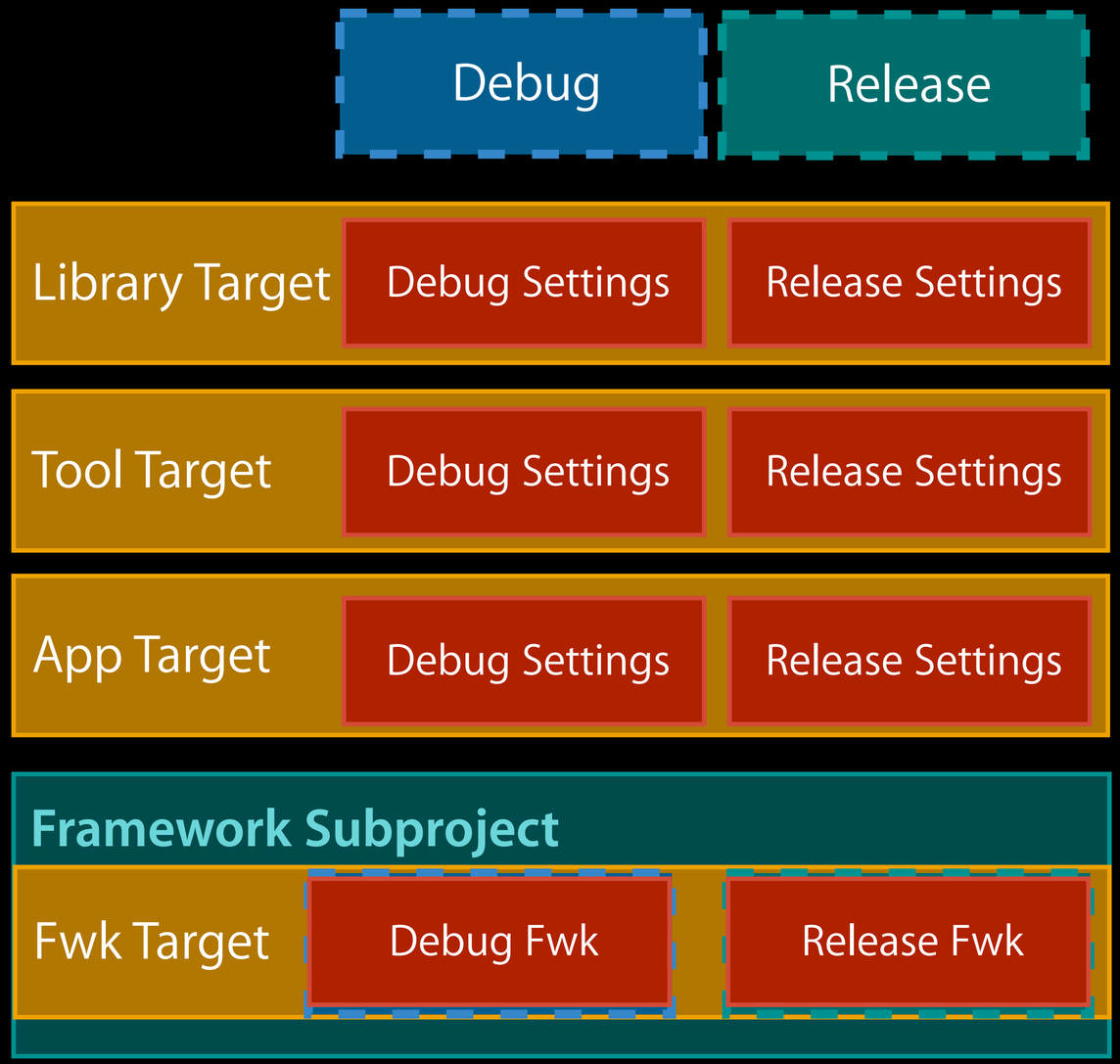
- Build styles imposed on all targets
- Style settings override target settings
- Style settings override subproject settings

Build Configurations

A matrix of settings, not a product



- Each target configuration is independent
- You can make all changes at the target level
- What you see in the target inspector is what you get at build time



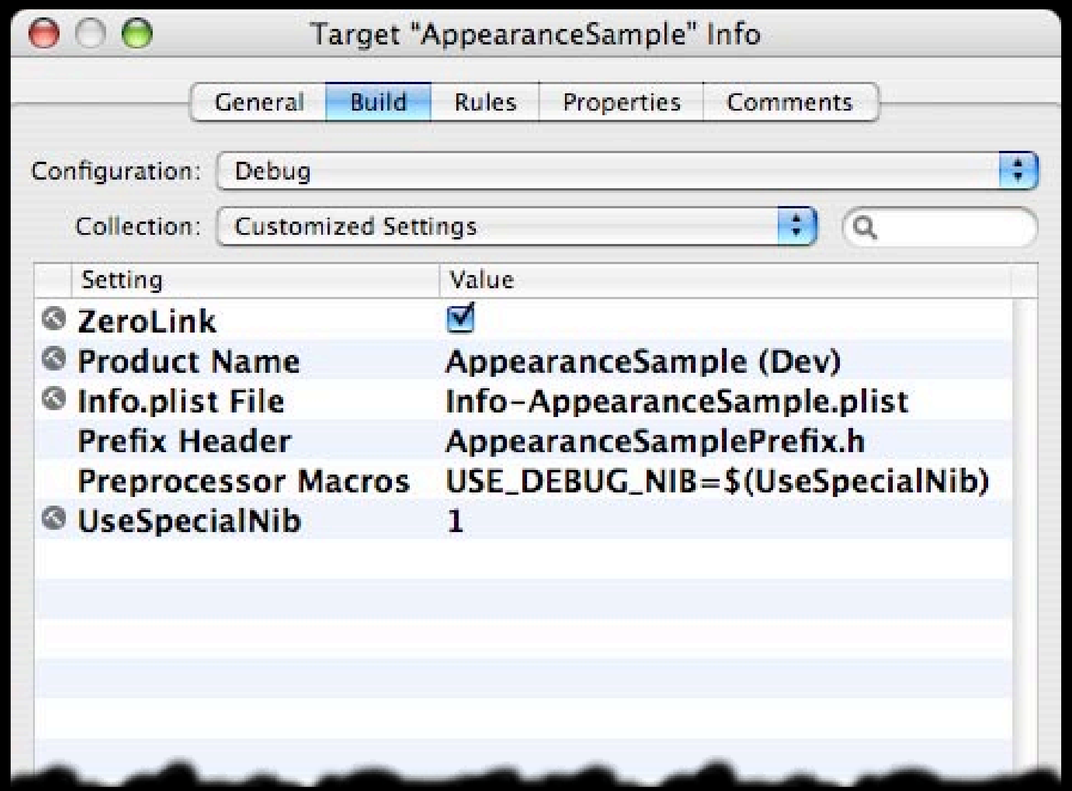
Build Configurations

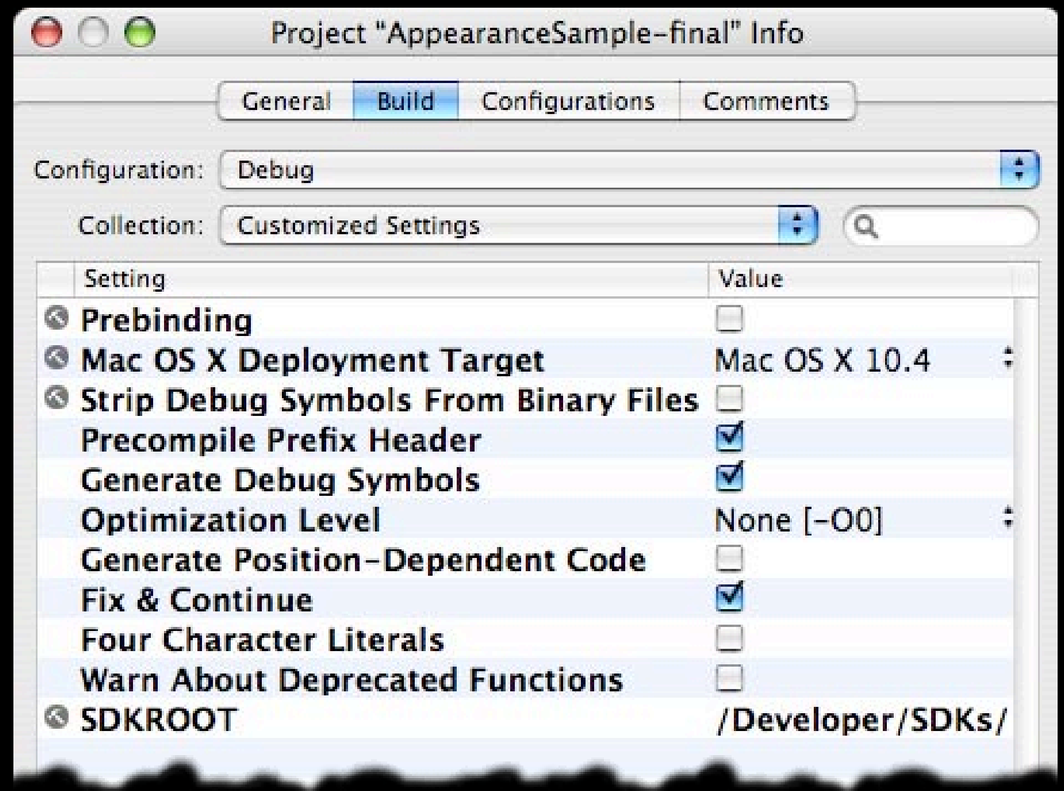
Pass by name, not by value

- Subprojects use the *corresponding* configuration, not the parent's settings themselves
- Can edit all configurations for given target at once
- Can edit one configuration for all targets (multiple select)

Project Build Configurations

- What if you *do* want the same settings across multiple targets?
- How do you make sure that new targets have the right default settings?
- How do you make target settings as concise as possible?





Configuration Files

```
ZERO_LINK = YES
UseSpecialNib = 1
PREBINDING = NO
SYMROOT = build
BUILD_VARIANTS = normal
LIBRARY_STYLE = STATIC
COPY_PHASE_STRIP = NO
GCC_OPTIMIZATION_LEVEL = 0
MACOSX_DEPLOYMENT_TARGET = 10.4
```

- .xcconfig text files in your project
- A list of SETTING = VALUE pairs
- You can prototype with drag and drop from inspectors
- Multiple targets can share the same config file
- You can reuse the config files across projects

Demo

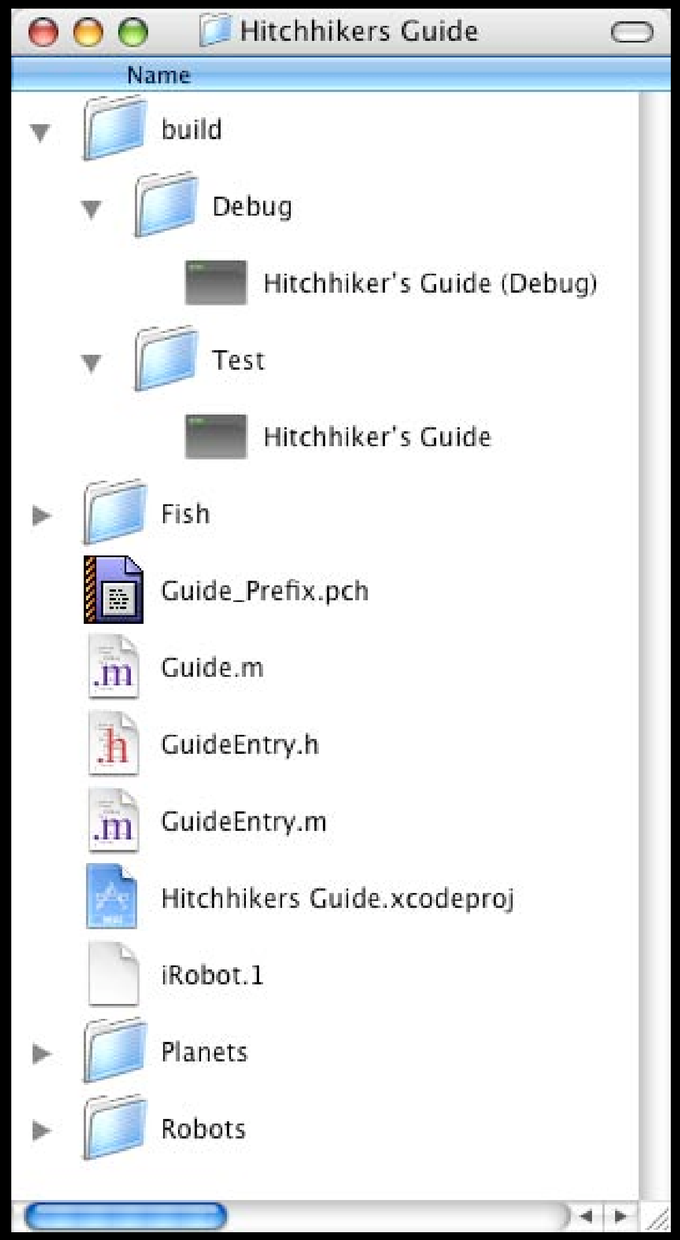
Build Configurations

Scott Tooker

http://developer.apple.com/wwdc2005/

Where Should a Setting Go?

Purpose of Setting	Best Location
Set project-wide standards (dialect, warnings)	.xcconfig file shared among project configurations
Standardize on styles of debugging and release (optimization, symbols, ZeroLink)	Per-configuration project settings
Characterize a particular target (product type, path, etc.)	.xcconfig file shared among a target's configurations
Distinguish among the debug and release flavors of a target (name, preprocessor flags)	Target configuration settings
Handle a special case with file-specific warning, optimization, or code-generation options	Per-file settings



Per-Configuration Build Folders

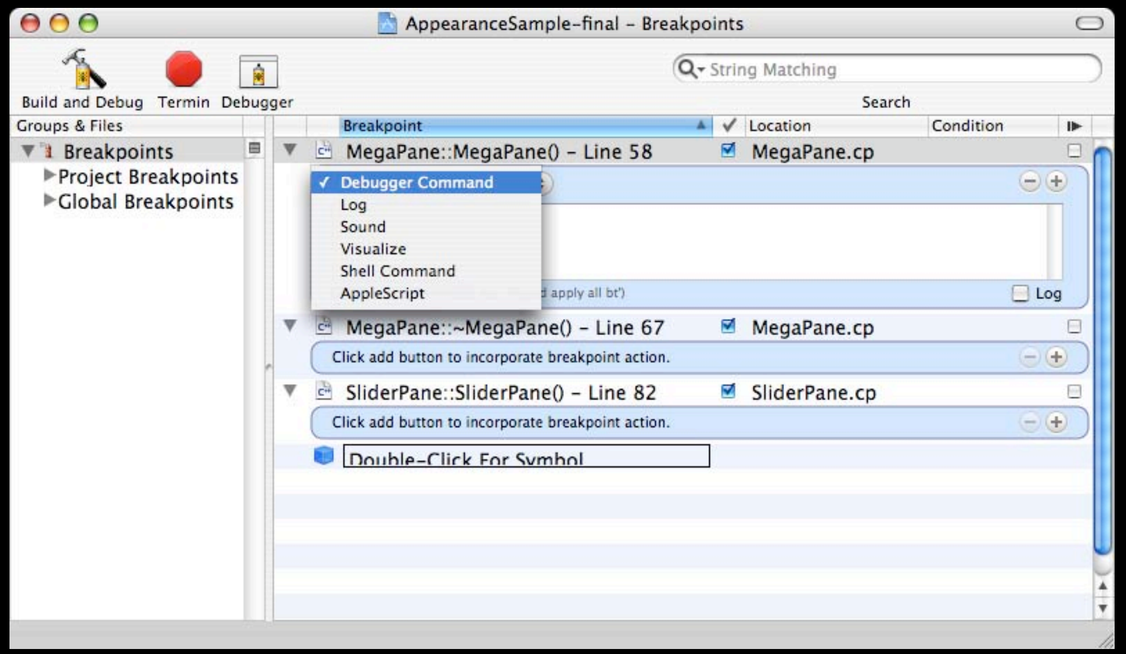
- Each configuration of each target has its own intermediates and products
- No need to rebuild just because you switched configurations

Build Configuration Summary

- We've eliminated the number one frustration about build settings
  "Why can't I see and set that here in the target inspector??!!"
- Configurations have more intuitive inheritance from project to target
- Configurations do the right thing with subprojects
- Project configurations and configuration files give a significant amount of expressive power
- Configurations help you manage the creation of universal binaries

Breakpoints

http://developer.apple.com/wwdc2005/

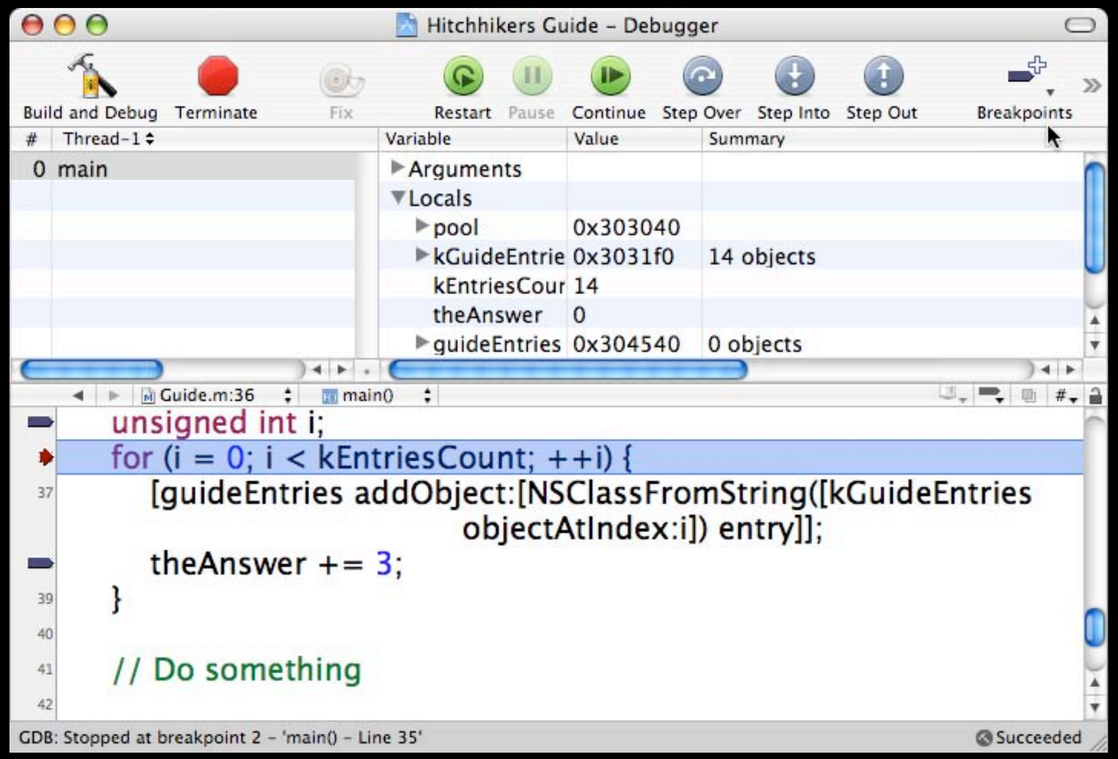


Breakpoints

- Conditions
- Actions
- Visualization
- Continuing after pause
- Watchpoints

Breakpoint Conditions

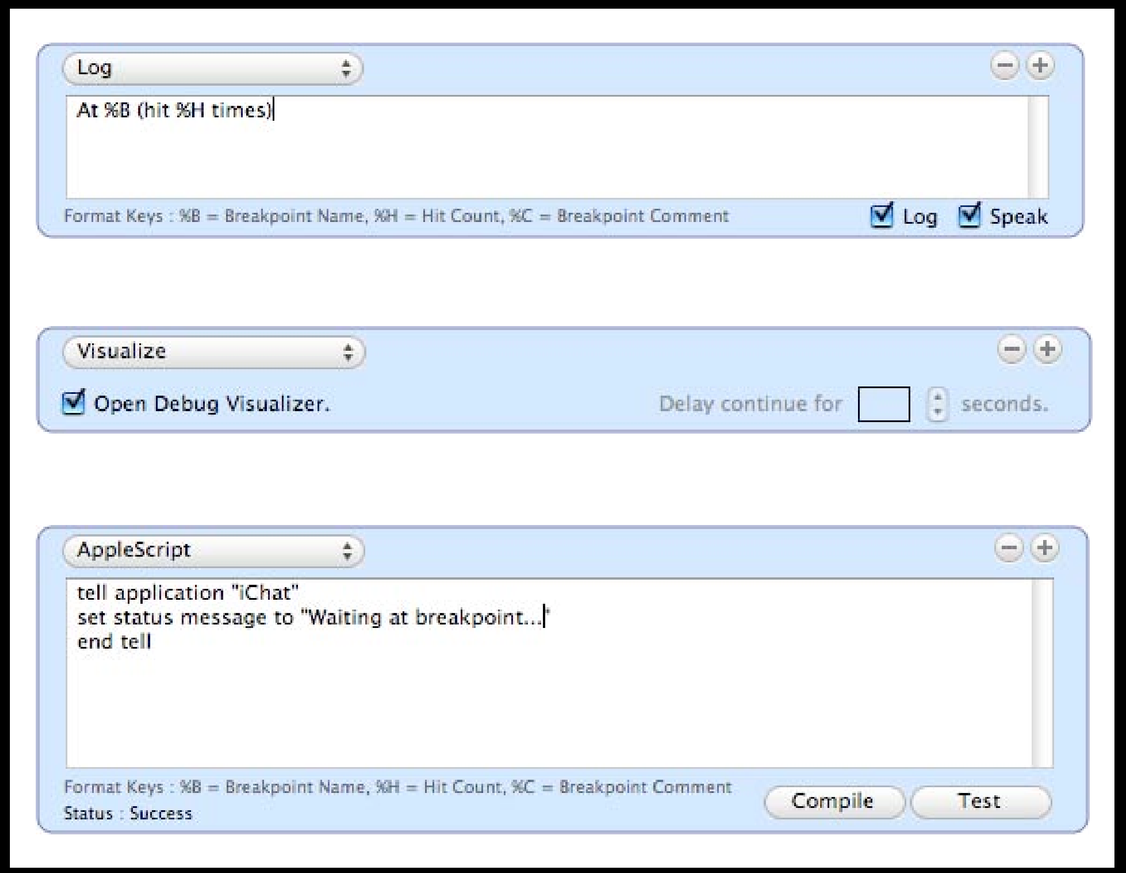
Any valid expression that could evaluate to non-zero



- You can use any variables in the current scope
- You can call functions in the target
  - You must cast the return types
  - It's best to only call simple, reliable, bounded functions at debug time
- You don't have access to preprocessor macros

Breakpoint Actions

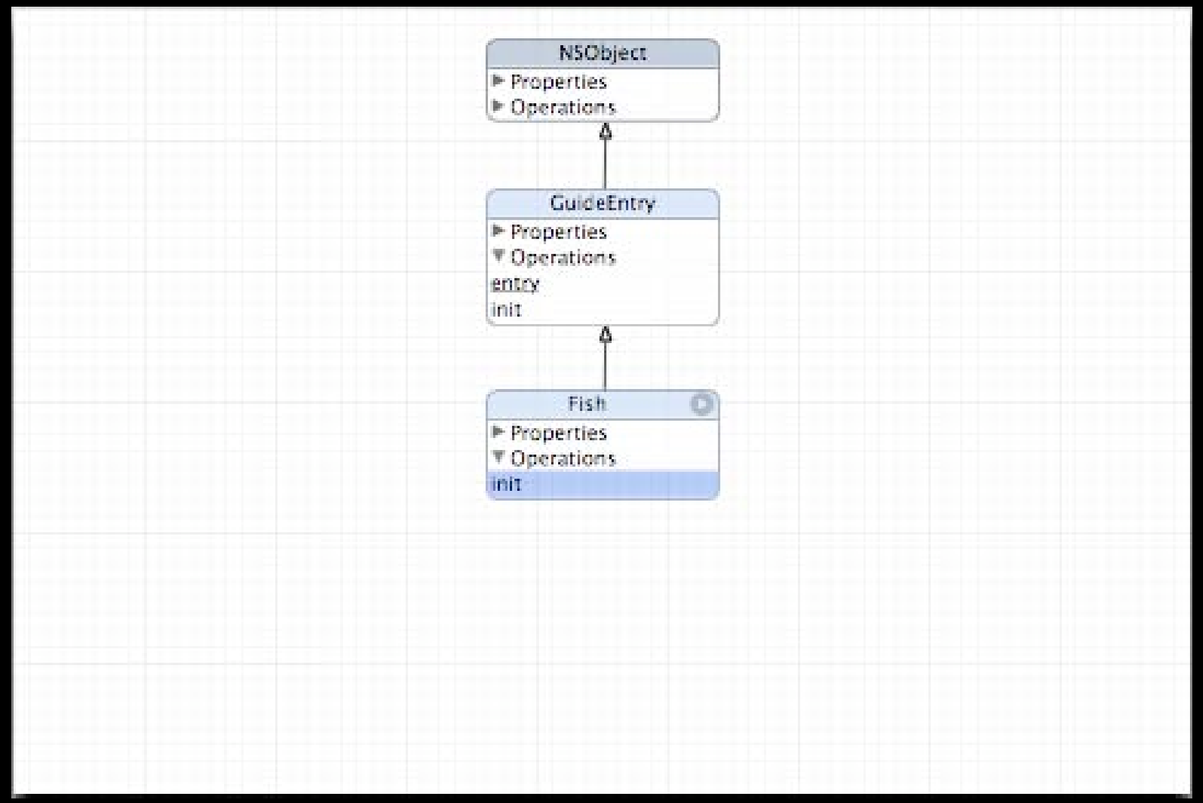
Can be performed at any breakpoint, normal or conditional



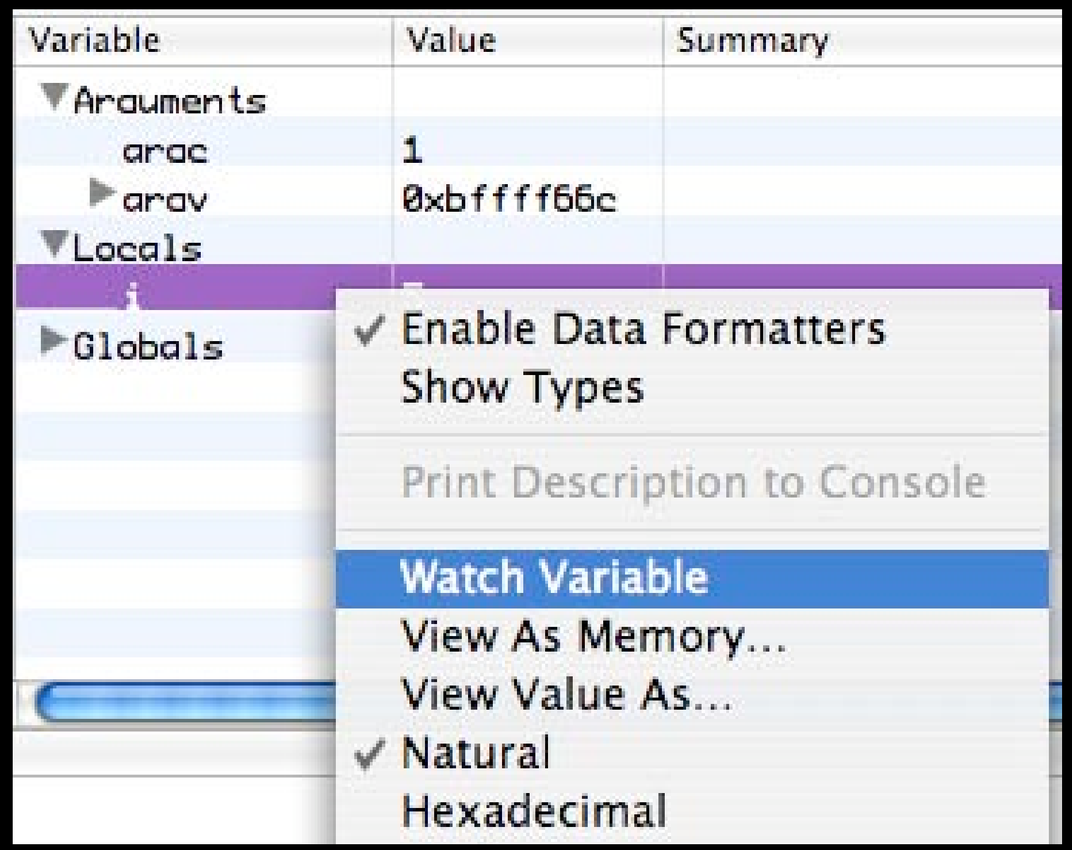
- Log and/or speak a message
- Play any specified sound
- Visualize the current class and method in a class diagram
- Execute any shell command
- Execute any AppleScript
- Execute any gdb command
  - If remote debugging, this is executed on the remote machine

The Visualizer

- Gives a dynamic overview of program flow
- Only for object-oriented code
- Shows classes, not instances



Watchpoints



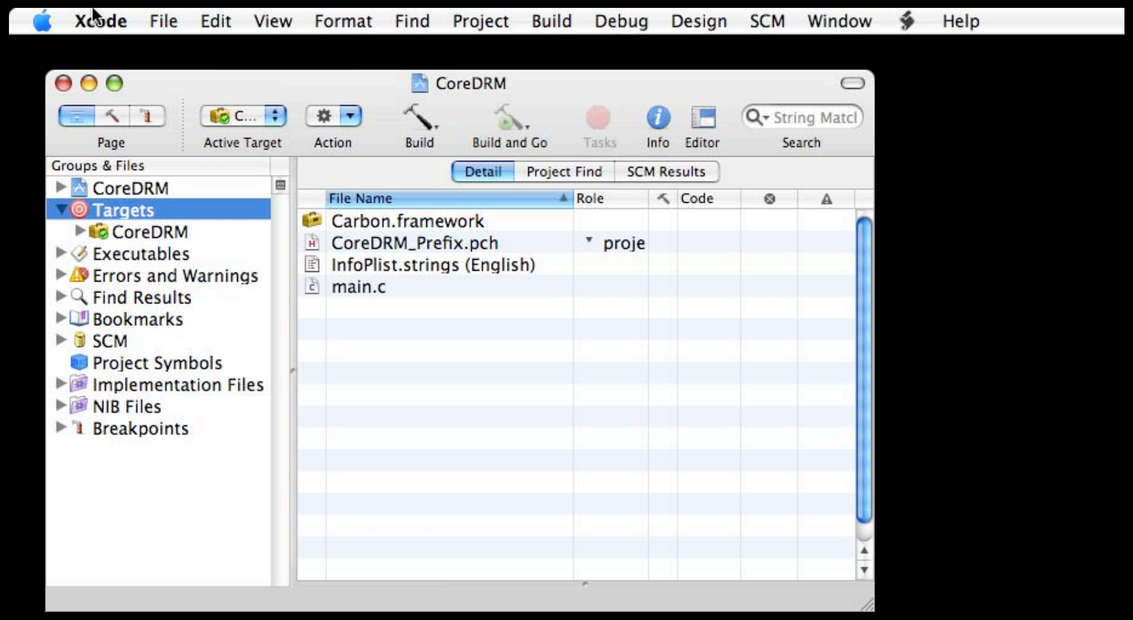
- Exposing gdb watchpoints to Xcode debugger
- Marks memory page as read only
  - Slower than hardware watchpoint
  - No size limitation
- Slows execution
  - Very slow, in some cases
- Last resort
- Not possible to watch values on the stack
  - e.g. most locals

Breakpoints Summary

http://developer.apple.com/wwdc2005/

Integrated Unit Testing

http://developer.apple.com/wwdc2005/



Unit Testing in Xcode

- A set of test harness tools for C-based languages
- A framework that test cases use for running and reporting
- A target template that builds and executes the test cases
- Test case results are logged to the Build Results window
- “Opt-in” model of testing

Unit Testing Demo

Chris Hanson

http://developer.apple.com/wwdc2005/

Unit Tests Summary

- A great development methodology
- Supported in Xcode 2.1

Project File Format Changes

http://developer.apple.com/wwdc2005/

Project File Format Changes

- Readability
  - Comments show names of objects being referenced
- Mergeability
  - Things that are atomic (file references) are all on one line
  - Things that are discrete (build settings) are on separate lines
- Writeability
  - UUIDs are optional and, if provided, are free-form (any unique string is OK)
  - Project file format is complete and definitive, no “import format” translation

Project File Format

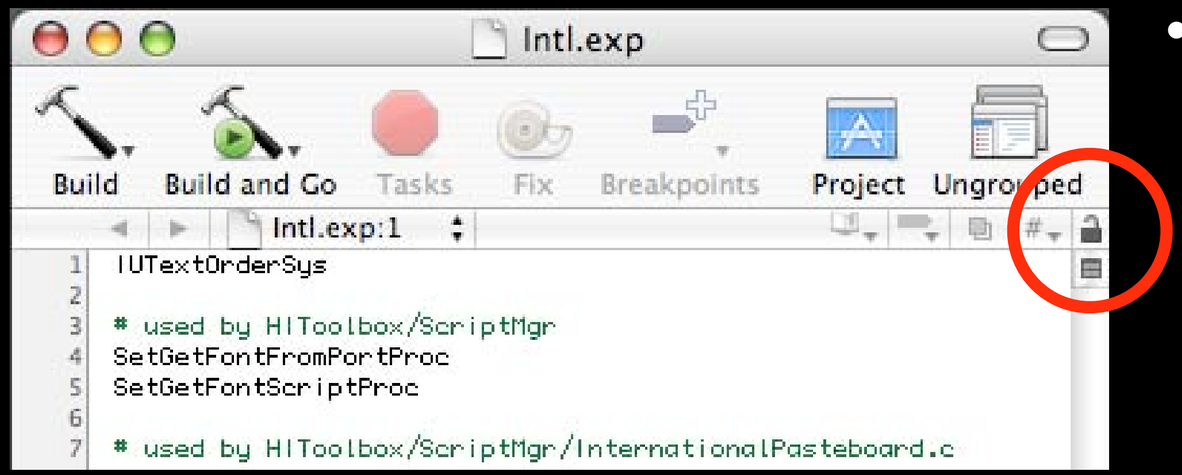
```
/* Begin XCBuildConfiguration section */
    502A915D0846F79F0090DB8D /* Release */ = {
        isa = XCBuildConfiguration;
        buildSettings = {
            COPY_PHASE_STRIP = YES;
            GCC_DEBUGGING_SYMBOLS = used;
            GCC_ENABLE_FIX_AND_CONTINUE = NO;
            GCC_GENERATE_DEBUGGING_SYMBOLS = NO;
            GCC_MODEL_TUNING = G3;
            GCC_OPTIMIZATION_LEVEL = 0;
            GCC_PRECOMPILE_PREFIX_HEADER = YES;
            GCC_PREFIX_HEADER = SDKExample_Prefix.pch;
            GCC_WARN_ABOUT_DEPRECATED_FUNCTIONS = NO;
            GCC_WARN_ABOUT_RETURN_TYPE = YES;
            GCC_WARN_UNUSED_VARIABLE = YES;
            INFOPLIST_FILE = Info.plist;
            INSTALL_PATH = "$(HOME)/Applications";
                        MACOSX_DEPLOYMENT_TARGET = 10.2;
                        PREBINDING = NO;
                        PRODUCT_NAME = SDKExample;
                                    WRAPPER_EXTENSION = app;
                        ZERO_LINK = NO;
                };
                name = Release;
        };
```

Other Xcode 2.1 Enhancements

http://developer.apple.com/wwdc2005/

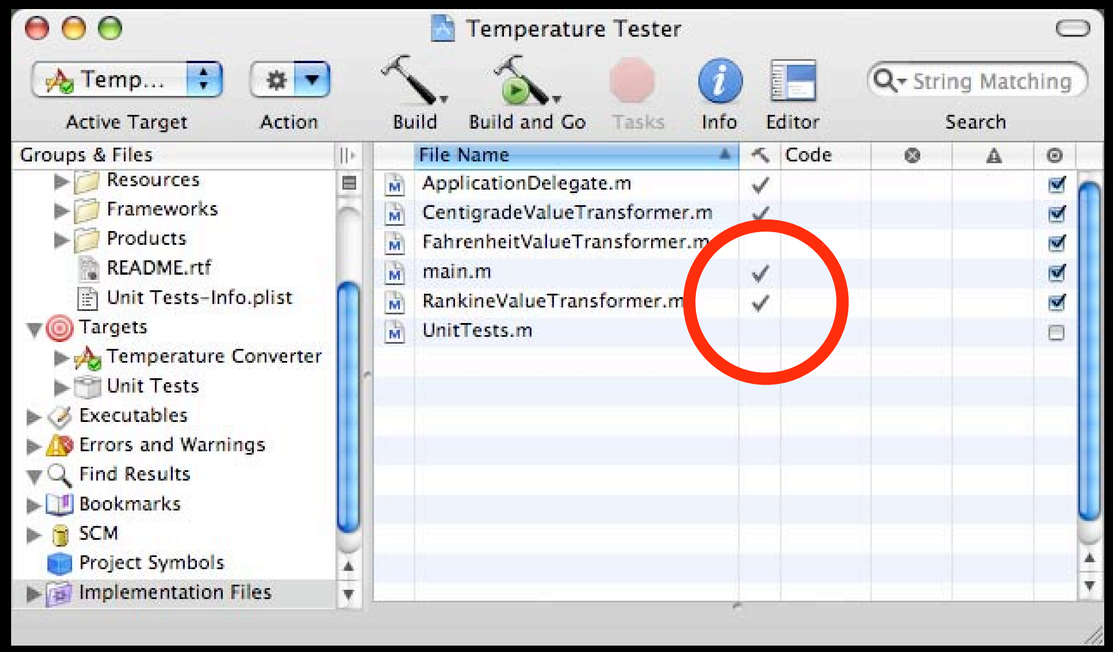
User Interface

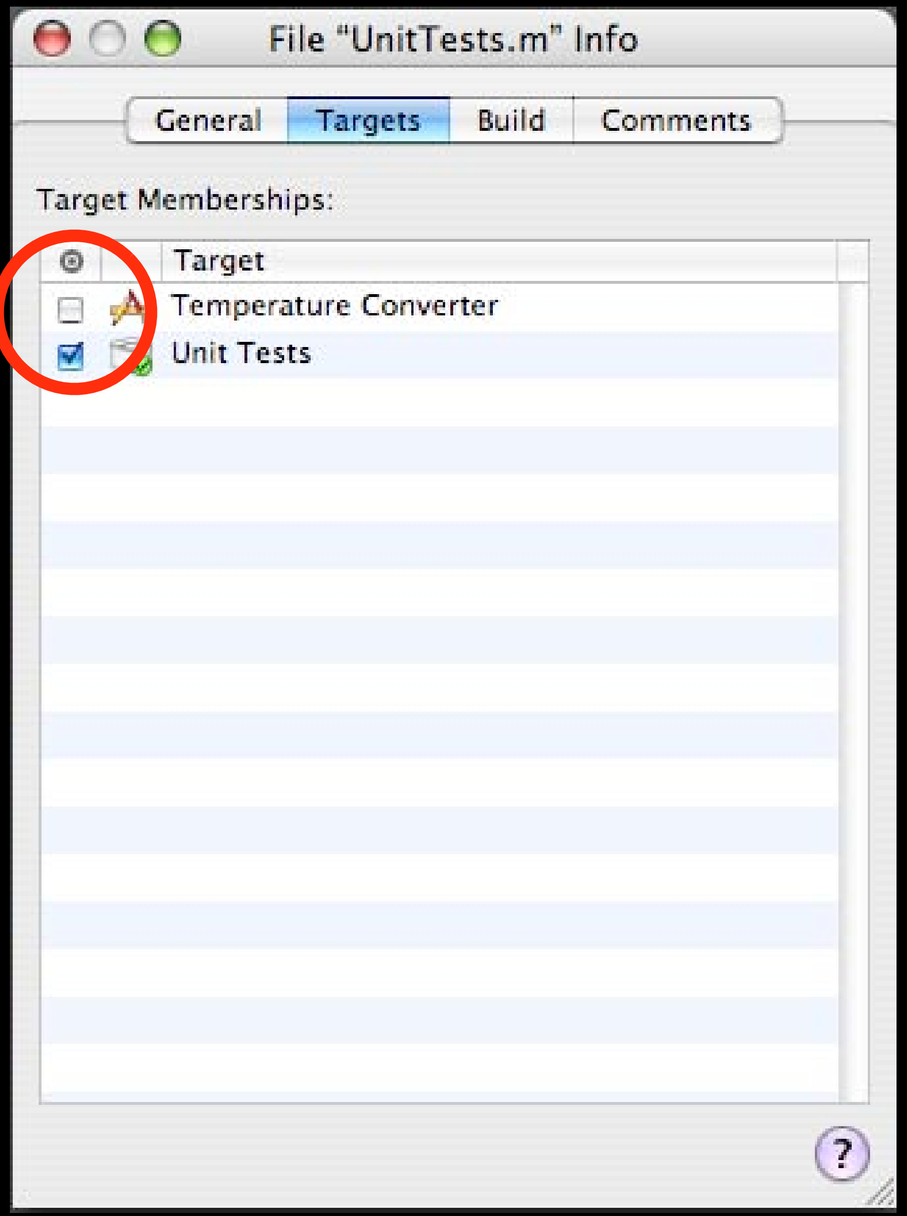
- Unlock file widget



User Interface

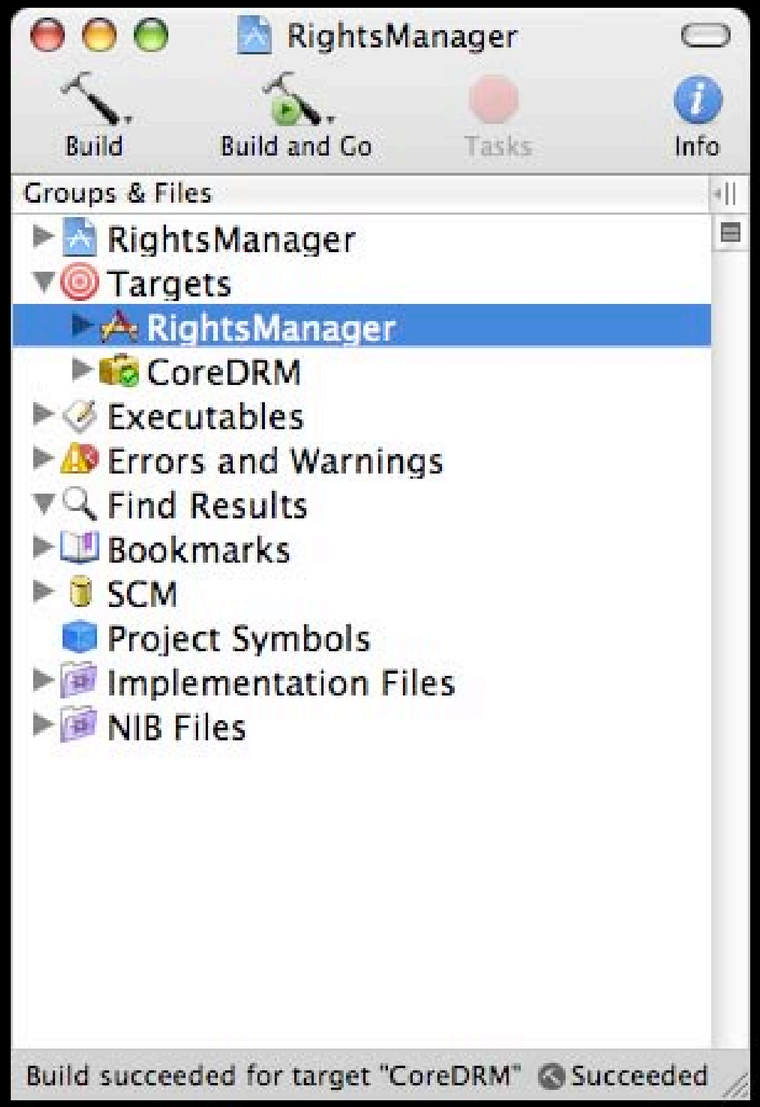
- Unlock file widget
- Clicking in build column “touches” file





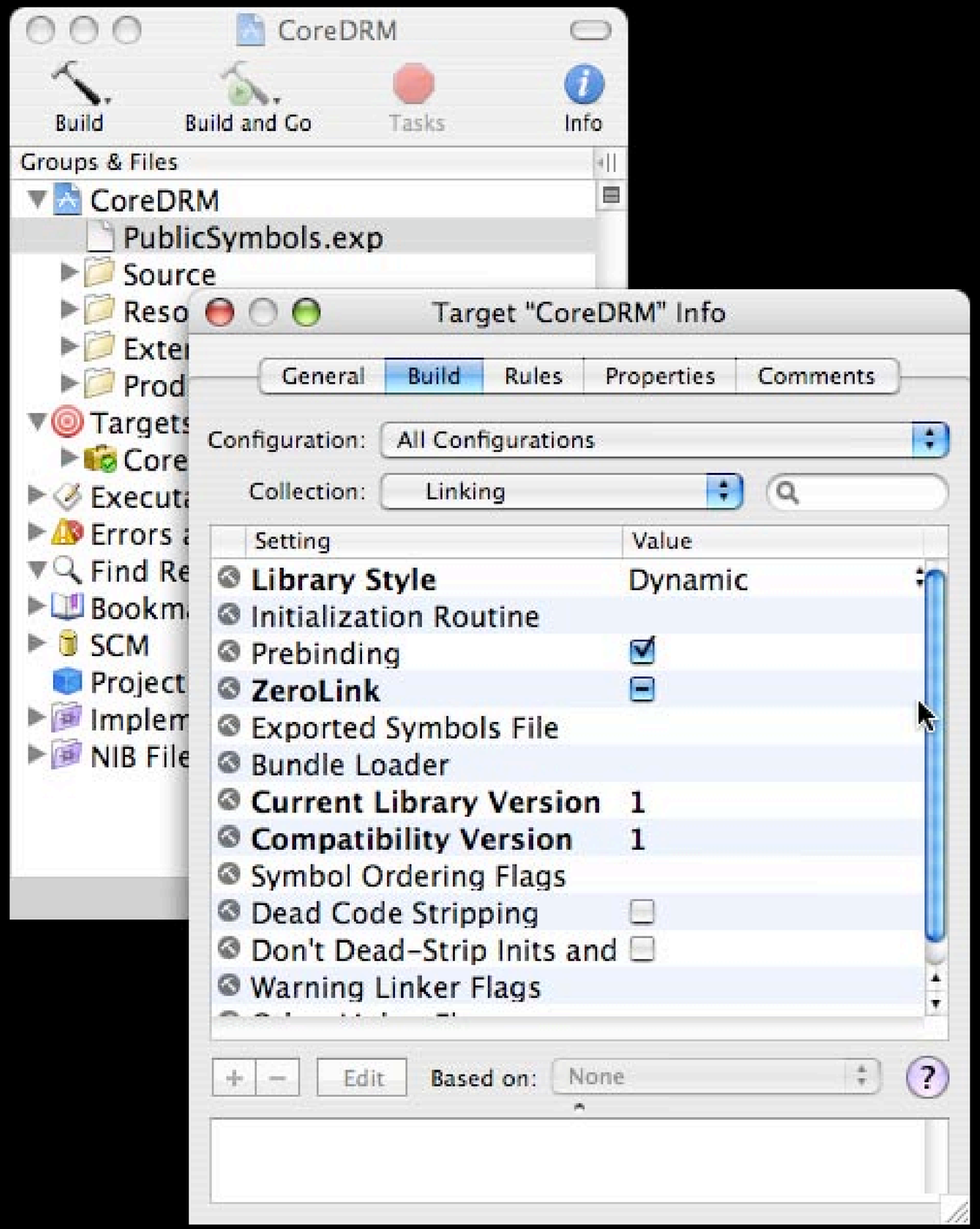
Inspectors

- Target membership tab in all file and group inspectors



Inspectors

- Target membership tab in all file and group inspectors
- More drag and drop:
  - Target dependencies



Inspectors

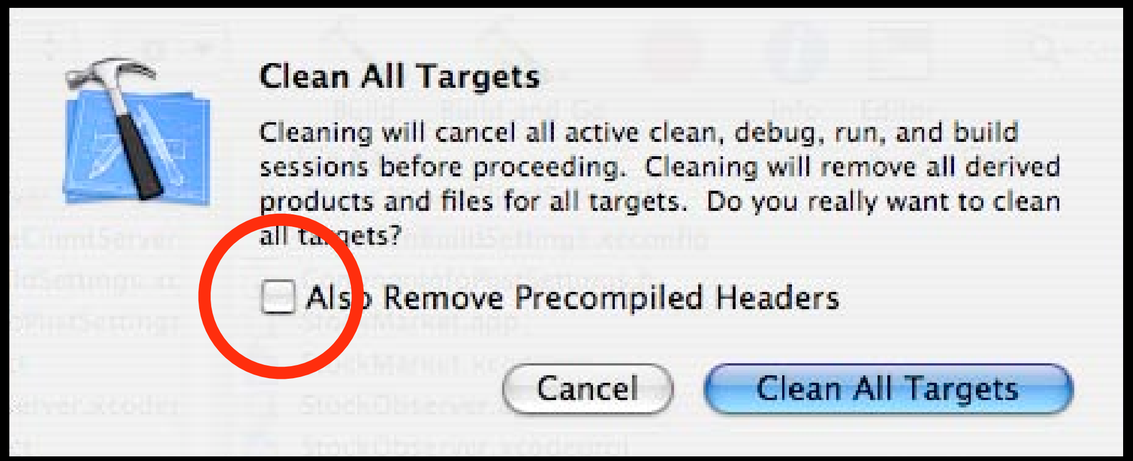
- Target membership tab in all file and group inspectors
- More drag and drop
  - Target dependencies
  - Build settings files and paths

Building

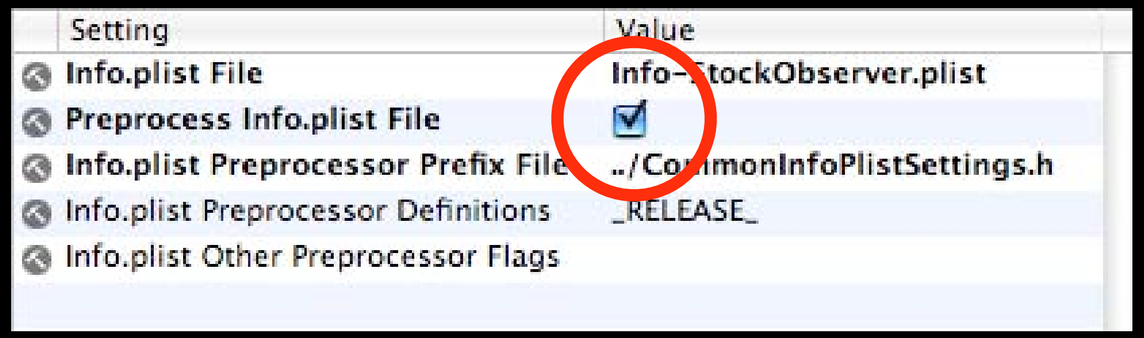
- Better dependency checking

Building

- Better dependency checking
- Clean without removing precompiled headers



Building

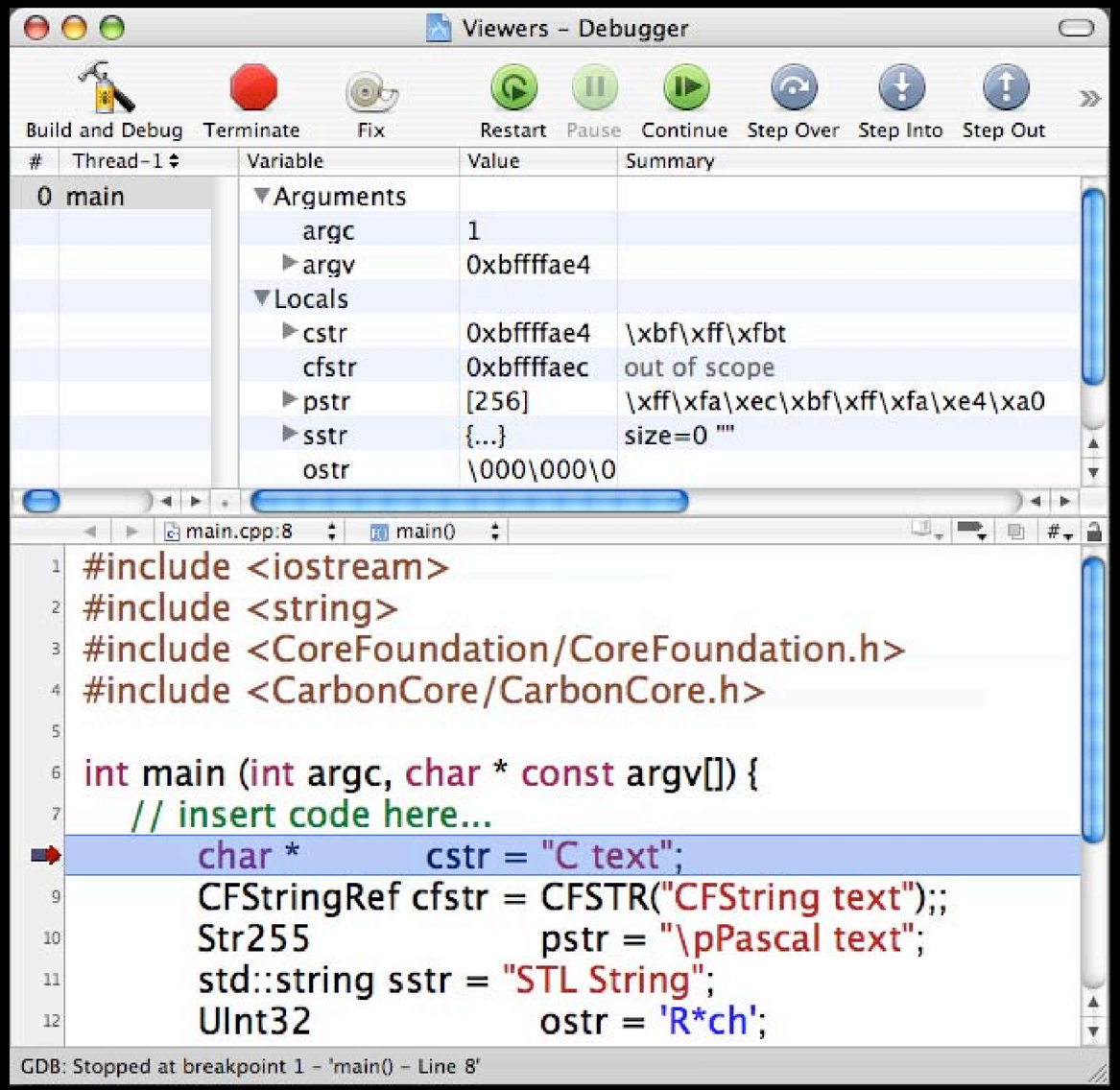


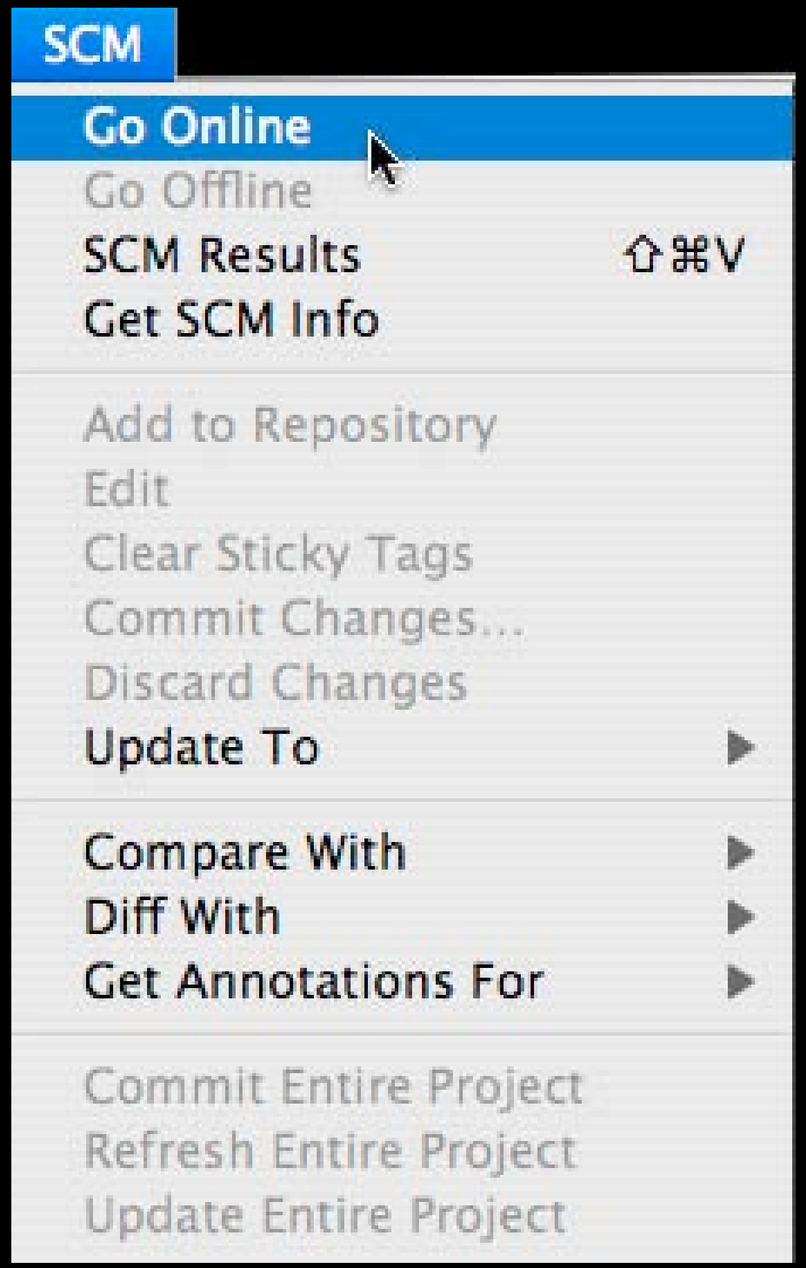
- Better dependency checking
- Clean without removing precompiled headers
- Preprocess Info.plist files

```
#ifdef _RELEASE_
 <string>Copyright (c) 2005</string>
#else
 <string>_CONFIGURATION_ version built on
__DATE__ __TIME__</string>
#endif
```

Debugging

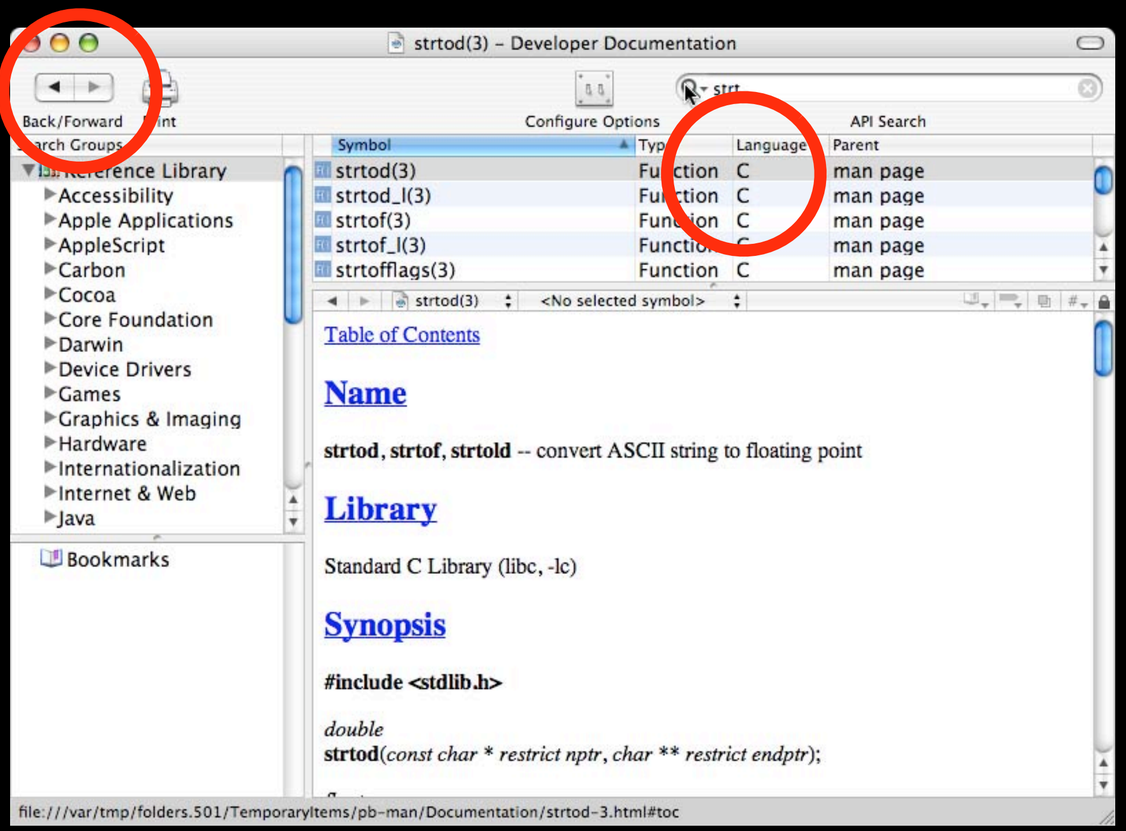
- String and OSType viewers
- std::string viewers
- Tab-completion in console





SCM

- Master online/offline control
- Support for unwrapped package files (nibs, models, RTFD, etc.)
- Improvements to subversion and per-force support

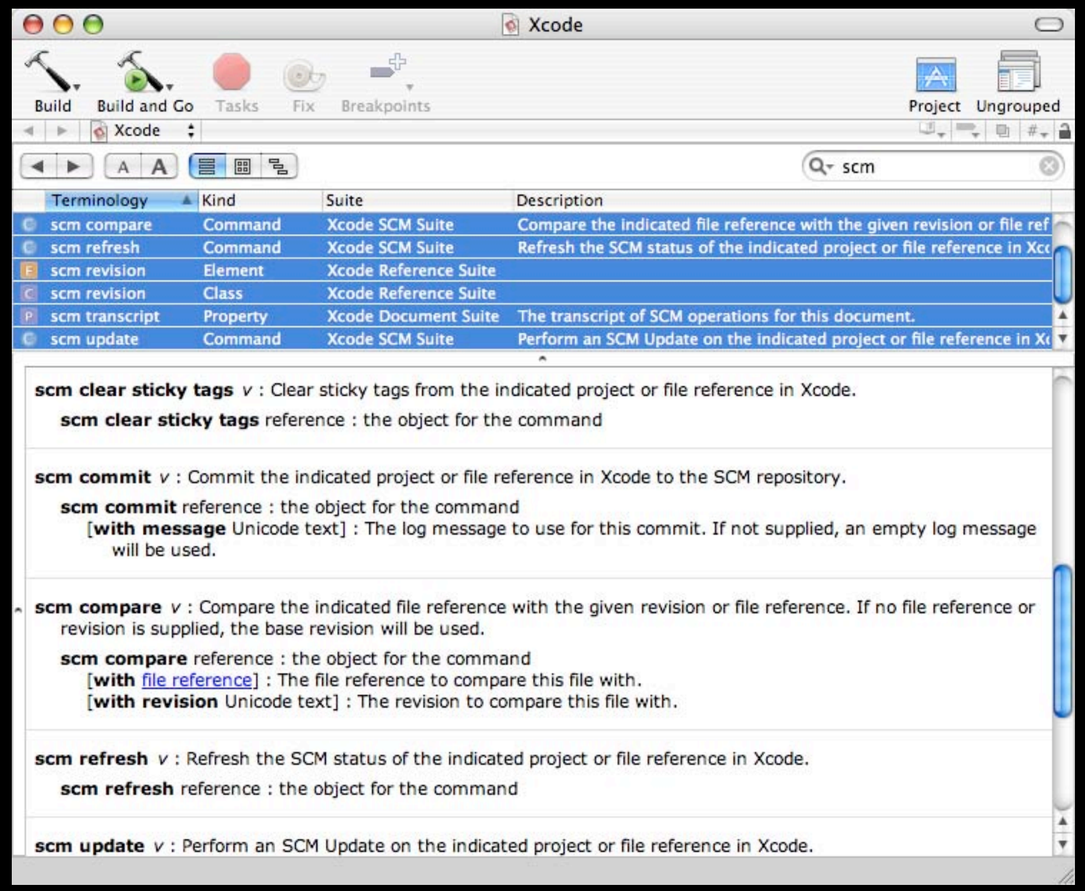


Documentation Viewer

- Standard C library is now indexed in the full-text and API indexes
- Man pages are now indexed in the full-text index
- Browser-like back/forward buttons

AppleScript

- Upgrade project
- SCM operations
- Build configurations
- Create targets



Summary: Xcode 2.1

http://developer.apple.com/wwdc2005/

Xcode 2.1

- Powerful, intuitive build settings management
- Versatile debugging
- The UI features you've been wanting
- Built-in support for universal binaries

More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Related Sessions

Adopting Xcode Hands-On	Session 402 Today 10:30AM
Power Programming with Xcode	Session 403 Wednesday 5:00PM
Project Management Mastery with Xcode	Session 408 Friday 2:00PM
Moving to gcc 4.0	Session 404 Wednesday 5:00PM

Related Labs and Feedback

Xcode Support	Lab 4000 Tuesday 2:00–5:00PM
Xcode Support	Lab 4001 Wednesday 9:00–12:00PM
Xcode Support	Lab 4002 Thursday 9:00–12:00PM
Developer Tools Feedback Forum	FF410 Friday 5:00PM

Whom to Contact

Matthew Formica
Developer Tools and Cocoa Evangelist
mformica@apple.com

xcode-Users Discussion List
Community-Based Support for Xcode Users
xcode-users@lists.apple.com

Q&A

Matthew Formica
Developer Tools and Cocoa Evangelist
mformica@apple.com

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference